

ИНТЕГРАЦИОННАЯ СРЕДА ГЕТЕРОГЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ

Обратимся к вопросам интеграции компонентов информационной системы на примере системы управления бизнес-процессами. Интеграционная среда (ИС) обеспечивает взаимодействие компонентов системы, а также осуществляет связь компонентов (служит мостом) с внешними системами. Необходимым требованием к устройству такой системы является поддержка открытых стандартов и спецификаций. Предлагается рассматривать интеграционную среду в двух плоскостях:

- ИС как инфраструктура обеспечения взаимодействия;
- ИС как набор сервисов, необходимых для функционирования бизнес-системы.

В данной работе ИС рассматривается в качестве инфраструктуры обеспечения взаимодействия.

Можно выделить следующие ключевые составляющие интеграционной среды открытой системы управления бизнес-процессами:

1. Подсистема управления процессами. Данный компонент отвечает за управление бизнес-процессами, описание которых составлено в соответствии со стандартом OASIS BPEL4WS. Каждый процесс порождает набор сервисов. Исполняемый процесс имеет, как минимум, один инициализационный сервис, при вызове которого создается новый экземпляр процесса. Каждый процесс может быть как потребителем, так и источником сервисов.
2. Среда-медиатор. В качестве такой среды используется ESB (Enterprise Service Bus) – сервисная шина, совместимая с открытым стандартом JBI (Java Business Integration). Сервисная шина позволяет организовать сопряжение компонентов гетерогенной информационной системы. Причиной гетерогенности является тот факт, что компоненты информационной системы взаимодействуют друг с другом посредством различных интерфейсов и транспортных протоколов. Благодаря тому, что подсистема управления процессами взаимодействует с компонентами системы через среду-медиатор, процессы имеют возможность потреблять весь спектр доступных в данной среде ресурсов, в том числе и те, которые, строго говоря, не являются сервисами (например, «обычные» Java-объекты).
3. Реестр сервисов – обеспечивает публикацию и обнаружение сервисов, входящих в состав информационной системы. Очевидно, что без использования реестра адреса (идентификаторы) всех используемых сервисов должны быть указаны в параметрах конфигурации, что требует постоянного обновления конфигурации экземпляров всех используемых продуктов. В качестве реестра сервисов предлагается использовать решение на основе стандарта OASIS UDDI. В данный момент

использование UDDI-реестра затруднено отсутствием актуальной открытой реализации стандарта UDDI. По этой причине на начальном этапе предлагается использовать динамическую маршрутизацию сервисов с помощью среды-медиатора.

Несмотря на то, что среда-медиатор предназначена для сопряжения разнородных интерфейсов и транспортных протоколов, представляется целесообразным проектировать интерфейсы компонентов информационной системы таким образом, чтобы взаимодействие осуществлялось через протокол SOAP поверх HTTP. Следование данному правилу позволяет избежать излишнего количества преобразований, а также дает возможность использовать набор протоколов WS-*.

Как было отмечено выше, в данный момент использование UDDI-реестра представляет определенные трудности, поэтому предлагается выбрать альтернативный способ динамического обнаружения сервисов. Всем компонентам, входящим в состав интеграционной среды, достаточно хранить сведения о размещении единственного сервиса-шлюза, созданного средой-медиатором. Доступ к имеющимся в системе сервисам производится не напрямую, а через шлюз.

Представим схему динамической маршрутизации сервисов через среду-медиатор:

1. Клиент формирует SOAP-запрос, предназначенный для удаленного сервиса.
2. Следуя стандарту WS-Addressing [4], клиент указывает в заголовке сообщения адрес сервиса-получателя. Данный адрес может не соответствовать реальному размещению удаленного сервиса.
3. Сформированное сообщение отправляется в шлюз, созданный средой-медиатором.
4. Среда-медиатор производит анализ адресного заголовка и, руководствуясь собственными правилами, определяет получателя сообщения.

Преимущества данного подхода:

- При изменении параметров размещения сервисов не требуется изменения конфигурационных файлов всех компонентов системы.
- В программном коде не требуется осуществлять явный поиск и публикацию сервисов в реестре.
- В существующем программном коде компонентов системы понадобится провести минимальные изменения, а именно, согласно стандарту WS-Addressing помещать в адресный заголовок уникальный идентификатор сервиса-получателя.
- В случаях, когда интерфейс какого-либо удаленного сервиса изменяется, с помощью среды-медиатора возможно производить автоматическую трансформацию передаваемых сообщений либо, что особенно важно, реализовать поддержку версионности сервисов. Например, если была реализована новая версия интерфейса какого-либо сервиса, но требуется

сохранить работоспособность существующих приложений, то среда-медиатор имеет возможность анализировать входящие SOAP-сообщения и на основе определенных признаков (таких как используемое пространство имен) перенаправить сообщение в адрес той или иной версии сервиса.

- В перспективе возможно реализовать централизованную аутентификацию сервисов посредством внедрения стандарта WS-Security.
- Средствами среды-медиатора возможно организовать автоматическое равномерное распределение нагрузки путем маршрутизации запросов к различным экземплярам сервиса в случае кластерной топологии системы.

Среди недостатков данного подхода следует отметить понижение надежности системы в целом из-за использования среды-медиатора в качестве центральной точки взаимодействия.